1.9

Interrupts

The VIC068A offers complete VMEbus and local bus interrupt generation and handling functions. In addition, the VIC068A also offers error and status interrupts for various VIC068A features. Local interrupt 2 (LIRQ2) may also be used as a periodic “heartbeat” timer. Significant control over the VIC068A interrupt generation/handling capabilities through the control registers listed below (32 of the 59 VIC068A control registers are for interrupt generation/handling):

- VMEbus Interrupter Interrupt Control Register (VIICR)
- VMEbus Interrupt Control Registers 1−7 (VICR1−7)
- DMA Status Interrupt Control Register (DMASICR)
- Local Interrupt Control Registers 1−7 (LICR1−7)
- ICGS Interrupt Control Register (ICGSICR)
- ICMS Interrupt Control Register (ICMSICR)
- Error Group Interrupt Control Register (EGICR)
- ICGS Interrupt Vector Base Register (ICGSIVBR)
- ICMS Interrupt Vector Base Register (ICMSIVBR)
- Local Interrupt Vector Base Register (LIVBR)
- Error Group Interrupt Vector Base Register (EGIVBR)
- VMEbus Interrupt Request/Status Register (VIRSR)
- VMEbus Interrupt Vector Base Registers 1−7 (VIVBR1−7)

1.9.1 VMEbus Interrupter

The VIRSR controls the assertion of the VMEbus interrupts. VIRSR[7:1] control the assertion (and deassertion if desired) of IRQ*7−1 signals respectively. VIRSR[0] enables the setting and clearing of these bits. To issue an interrupt, both the interrupt bit and VIRSR[0] must be set. To clear an interrupt, the interrupt bit must be set and VIRSR[0] must be cleared. VIRSR[7:1] may also be read to indicate the status of pending VMEbus interrupts. For example, if *vic* contains the base address of the VIC068A registers, the following 68K code could be used to assert IRQ4*:

```
move.b #$11,(vic,$83)
```

This code would clear the interrupt:

```
move.b #$10,(vic,$83)
```

Once the interrupt is issued, the handler for that interrupt proceeds with the interrupt acknowledge cycle. Once the VIC068A recognizes a valid interrupt acknowledge cycle (IACKIN* asserted), it places the status/ID vector, located in the VIVBRi, on the D[7:0] lines. The VIC068A is capable of issuing 8-bit status/IDs only. The VIC068A uses a Release-On-AcKnowledge (ROAK) for the normal deassertion of the IRQi* signals.

More than one VMEbus interrupt may be issued or pending simultaneously by the same VIC068A interrupter.

1.9.2 The VIC068A VMEbus Interrupt Handler

The VIC068A may be enabled to handle VMEbus interrupts. If VICR1−7[7] is clear, the VIC068A handles all pending interrupts on that respective level. Only one VMEbus module should be configured to handle any given interrupt level per VMEbus system.

The VIC068A performs D8 interrupt acknowledge cycles only.

When the VIC068A detects a pending VMEbus interrupt, it performs the following functions:

1. Assert the IPLi signals with the value programmed in the VICRi.
2. Wait for the assertion of the FCIACK* signal, which indicates the local processor is acknowledging a local interrupt.
3. Once FCIACK* is asserted, sample LA[3:1] to determine the IPL value of local interrupt being acknowledged.
4. If matched, obtain VMEbus mastership, assert IACK*, drive A[3:1]=interrupt-level and enable D[7:0] to the local data bus LD[7:0].
5. Once DTACK* is asserted by the VMEbus interrupter indicating the status/ID byte is valid on D[7:0], the VIC068A asserts DSACKi* to complete the local interrupt acknowledge cycle and indicates that the status/ID is available on LD[7:0].

The LA[3:1] matching described in step 3 is discussed in section 1.9.3.

Table 1−14 summarizes the VMEbus interrupt acknowledge cycle.

Table 1−14. VMEbus Interrupt Acknowledge Cycle

Step	VIC068A Interrupter	VIC068A Handler	Local Processor
1	Generate IRQi* (write to VIRSR)		
2		Detect IRQi* asserted (VICRi enabled to handle interrupt)	
3		Assert inverted version of IPLi value as programmed in the VICRi	
4			Detect IPLi asserted
5			Place inverted IPL level of interrupt being handled on LA[3:1]
6			Assert FCIACK*, PAS*, DS*
7		Detect FCIACK* asserted	
8		Sample LA[3:1] for request level being acknowledged	
9		If matched, obtain VMEbus mastership	
10		Assert IACK* and place number of IRQ being acknowledged on A[3:1]	
11		Enable D[7:0] to capture status/ID vector	
12	Detect IACKiIN* asserted and sample A[3:1]		
13	If A[3:1] matches IRQ number requested, drive status/ID vector programmed in the VIVBRi		
14	Assert DTACK*		
15	Generate interrupt acknowledged interrupt if enabled in the VIICR	Assert DSACKi*	

Table 1−14. VMEbus Interrupt Acknowledge Cycle (continued)

Step	VIC068A Interrupter	VIC068A Handler	Local Processor
16			Capture status/ID
17			Enter Interrupt Service Routine

1.9.3 Local Interrupt Handler

The VIC068A may be enabled to handle local interrupts in addition to VMEbus interrupts. Local interrupt handling is enabled through the LICRs 1−7. If bit 7 is cleared in any of these registers, the corresponding local interrupt is handled by the VIC068A. The local interrupts may be configured in a variety of ways including edge or level sensitivity, active High/Low levels, or rising/falling edges. The VIC068A may be configured to supply, or autovector, the status/ID. If VIC068A is configured to supply the status/ID vector, the vector is supplied from the LIVBR. Unlike the VIVBRs, this is a single register. The upper 5 bits are user-defined, and the lower 3 bits are dynamic to indicate the interrupt value (LIRQ7*...LIRQ1*, etc.). These lower 3 bits are place-holders only. They may not indicate the interrupt value if read, even during an interrupt acknowledge cycle. If the VIC068A is configured for autovectoring (VIC068A does not supply status/ID), the VIC068A asserts the LIACKO* signal after the interrupt is acknowledged (FCIACK* asserted) by the local processor. This should indicate to the interrupter to place the status/ID on D[7:0] signal lines and assert DSACKi*. LIACK0* may be connected to the 68K AVEC signal for internal generation of the interrupt vector.

Once the VIC068A detects LIRQi* asserted, and the VIC068A is enabled to handle that interrupt, the VIC068A proceeds as follows:

1. Assert the IPLi signals with the value programmed in the LICRi.

2. Wait for the assertion of the FCIACK* signal, which indicates the local processor is acknowledging a local interrupt.

3. Once FCIACK* is asserted, sample LA[3:1] to determine the IPL value of local interrupt being acknowledged.

4. If matched,
   - drive LD[7:0] with status/ID if enabled to supply vector and assert DSACKi*.
   - or, assert LIACK0* if not enabled to supply vector (autovectoring).

The LA[3:1] matching described in step 3 is required to distinguish an acknowledge from among multiple pending interrupts. If the value of LA[3:1] does not match the value of the IPL signals (see section 1.9.8 for positive/negative logic issues regarding the IPL signals), the VIC068A assumes the acknowledge is not for it. The 68K family of processors asserts LA[3:1], as described above, automatically. *Table 1−15* summarizes the local interrupt acknowledge cycle.

Table 1−15. Local Interrupt Acknowledge Cycle

Step	Local Interrupter	VIC068A Handler	Local Processor
1	Generate LIRQi*		
2		Assert inverted version of IPLi value programmed in the LICRi	
3			Detect IPLi asserted
4			Place inverted IPL level of interrupt being handled on LA[3:1]
5			Assert FCIACK*, PAS*, DS*
6		Detect FCIACK* asserted	
7		Sample LA[3:1] for request level being acknowledged	
8a		If matched, drive status/ID on LD[7:0] programmed in the LICRi	
9a		Assert DSACKi*	
8b		Assert LIACK0*	
9b	Drive status/ID, DSACKi*		
10			Capture status/ID
11			Enter ISR

1.9.4 The FCIACK Cycle

When the VIC068A detects an interrupt (either VME or local) that it has been programmed to handle, the VIC068A begins a FCIACK cycle. The IPL value associated with that interrupt (as programmed in VIICR, VICR1−7, DMASICR, LICR1−7, ICGSICR and ICMSICR) is placed into an internal lookup table and driven (inverted) onto the IPL

lines. The value placed onto LA[3:1] by the local processor during a FCIACK cycle is compared with the lookup table within the VIC068A. If a match is found, the interrupt that the local processor has agreed to service is handled.

Since interrupts occur asynchronous to each other, it is possible for the value of the IPL lines to change without warning. If the IPL lines were not sampled prior to them changing, that interrupt will not be seen by the local processor until all higher priority interrupts have been handled. Once FCIACK* is asserted to the VIC068A, the IPLi lines will be frozen until the Interrupt Acknowledge cycle is complete. Since the value placed onto LA[3:1] by the local processor is compared against a lookup table and not the current value of the IPL lines the VIC068A will handle any pending interrupt that the local processor agrees to handle during a FCIACK cycle.

1.9.5 The Error/Status Interrupts

The VIC068A is capable of generating local interrupts on certain error or status conditions. The error group interrupts include:

- SYSFAIL* assertion. An interrupt is generated when SYSFAIL* is detected by something other than the VIC068A.
- ACFAIL* assertion. An interrupt is generated when ACFAIL* is detected.
- Write post failure. An interrupt is generated when a master write post has failed due to a LBERR* or BERR* assertion.
- VMEbus arbitration failure. An interrupt is generated if the VIC068A is system controller and the VMEbus arbitration timeout timer expires.

The IPL value asserted for these error group interrupts is contained in the EGICR. There is only one IPL value for the error group interrupts.

The VIC068A contains two status interrupts:

- The DMA complete interrupt. An interrupt is generated when a block transfer with local DMA is complete (successful or unsuccessful). The IPL value issued is also contained in the DMAICSR.
- The Interrupter interrupt acknowledge. An interrupt is generated upon the acknowledgment of a previously issued VMEbus interrupt. The IPL value issued is contained in the VMEIICR.

Upon local acknowledgment of both the error and status group interrupts, the VIC068A issues the status/ID byte located in the EGIVBR. EGIVBR[7:3] are user defined. EGIVBR[2:0] are dynamic to indicate the particular interrupt being acknowledged.

1.9.6 Interrupt Priority Order

The 29 interrupt sources of the VIC068A are grouped into 19 priority categories. With multiple interrupts pending, the VIC068A issues the interrupts in the following order:

Priority	*Interrupt*
1	LIRQ7*
2	Error Group
3	LIRQ6*
4	LIRQ5*
5	LIRQ4*
6	LIRQ3*
7	LIRQ2*
8	LIRQ1*
9	ICMS Group
10	ICGS Group
11	IRQ7*
12	IRQ6*
13	IRQ5*
14	IRQ4*
15	IRQ3*
16	IRQ2*
17	IRQ1*
18	DMA Complete
19	VMEbus Interrupter

Notice that the priority of the interrupts within the VIC068A is not dependent on the IPL values programmed in the interrupt control registers. This, combined with the fact that VMEbus interrupt priority is also governed by the position of the module within a VMEbus system, makes determining actual interrupt priority for a given processor in a given VMEbus system flexible.

If an interrupt is pending, and another higher-priority interrupt (according to the above table) is issued to the VIC068A, the VIC068A will change the state of the IPL lines to that of the higher-priority interrupt. The VIC068A, however, still handles the lower-priority interrupt if it is acknowledged before the higher one. If the lower-priority interrupt is being acknowledged (FCIACK* asserted) at the time of the assertion of the higher-priority interrupt, the IPL signals will not change until the interrupt acknowledge cycle is complete.

While an interrupt is pending, all lower-priority interrupts subsequently issued are queued within the VIC068A and issued at the completion of the pending interrupt's acknowledge cycle.

1.9.7 Clock-Tick Interrupt Generator

LIRQ2 may be enabled to function as a "heartbeat" interrupt generator issuing periodic interrupts. If the interrupt generator is enabled by configuring SS0CR0[7,6], the LIRQ2* pin is converted to an output and issues periodic interrupts in 50-, 100-, or 1000-Hz frequencies. This interrupt should be considered an edge-triggered interrupt since it may be deasserted before the interrupt is acknowledged. The VIC068A may also be configured to handle the interrupt by enabling LIRQ2* in LICR2. In this case, the interrupt is considered to be like any other interrupt issued to the VIC068A and normal interrupt acknowledge procedures apply.

1.9.8 Interrupt Control Registers

The IPL values programmed in the interrupt control registers are positive logic values. This is unlike the value of the IPL signals asserted to the processor by the VIC068A, which are negative logic. The VIC068A performs this complementing of values internally. That is, if a value of 010 (positive logic 2) is programmed into an interrupt control register, the value of 101 (negative logic 2) is driven on the IPL signals.

When local resources are acknowledging an interrupt and driving LA[3:1] with the level of the interrupt being acknowledged, LA[3:1] should contain the positive logic value of the interrupt.

Mask bits exist for every interrupt the VIC068A is capable of generating. These mask bits are set (interrupts disabled) after a global reset. When changing the level and/or polarities of the local interrupts, it is recommended that the interrupts be masked prior to this and re-enabled after.